

Title of the Prior Art

Japanese Published Patent Application No. Sho.60-207967

Date of Publication : October 19, 1985

Inventor : Kunihiko Sakata

Concise Statement of Relevancy

Disclosed in this reference that a mask bit string is initialized in a shift register to provide a vector arithmetic processor which can apply an arithmetic pipeline processing in executing a masking vector arithmetic. Thus, it is required to perform the initialization before the conditional arithmetic is started.

On the other hand, the present invention decides the state of source data which are supplied in the pipeline, and performs a conditional arithmetic depending on the state of the source data as the condition, whereby it is not required to perform initialization before the conditional arithmetic is started.

This Page Blank (uspto)

⑬ 公開特許公報(A)

昭60-207967

⑤ Int. Cl.⁴
G 06 F 15/347識別記号 庁内整理番号
7056-5B

④ 公開 昭和60年(1985)10月19日

審査請求 未請求 発明の数 1 (全6頁)

⑭ 発明の名称 ベクトル演算処理装置

① 特 願 昭59-63563

② 出 願 昭59(1984)3月31日

⑦ 発 明 者 坂 田 邦 彦 東京都府中市東芝町1番地 東京芝浦電気株式会社府中工場内

⑧ 出 願 人 株 式 会 社 東 芝 川崎市幸区堀川町72番地

⑨ 代 理 人 弁 理 士 鈴 江 武 彦 外2名

明 細 書

1. 発明の名称

ベクトル演算処理装置

2. 特許請求の範囲

第1のアレイ・オペランド・データと第2のアレイ・オペランド・データ間の対応する要素毎の演算を、 n 段のパイプライン処理により行なう演算部を備えたマイクロプログラム制御方式のベクトル演算処理装置において、マスクビット列が初期設定され、当該マスクビット列を上記演算部のパイプライン処理に同期して1ビットずつ順次シフトするシフトレジスタと、上記第1のアレイ・オペランド・データの各要素を上記演算部のパイプライン処理に同期して1要素ずつ順次保持し、その都度その保持要素を次段に出力するバッファレジスタと、このバッファレジスタおよび上記シフトレジスタにそれぞれ接続され、上記バッファレジスタからの上記要素および上記シフトレジスタの所定位置からのマスクビットを、上記演算部のパイプ

ライン処理に同期して順次保持転送するパイプラインレジスタ群と、このパイプラインレジスタ群の最終段からの出力データ中の上記要素、または上記演算部の演算結果のいずれか一方を、マスク付ベクトル演算の演算結果の1要素として、上記パイプラインレジスタ群の最終段からの出力データ中の上記マスクビットに応じて選択出力する手段とを具備することを特徴とするベクトル演算処理装置。

3. 発明の詳細な説明

〔発明の技術分野〕

この発明は、マスク付ベクトル演算機能を有するベクトル演算処理装置に関する。

〔発明の技術的背景とその問題点〕

大衆のアレイ・オペランド・データを処理するベクトル演算の一つとして、マスク付ベクトル演算が知られている。このマスク付ベクトル演算は、オペランド・データに対して或るビット列(マスクビット列)のビット(マスクビット)をそれぞれ割当て、このビットに応じ、演

算を実行するか、或いはオペランド・データをそのまま出力するかを制御する演算である。

第1図は、マスク付ベクトル演算の一般的なフローチャートを示す。マスク付ベクトル演算では、まずビット列の中からオペランド・データに対応するビットが読出される。そして、このビットを分岐条件にして分岐し、演算を実行するか、或いは演算を実行せずにオペランド・データを出力し、次のデータの処理に移る。

ところで、ベクトル演算処理装置は、アレイ・オペランド・データを高速に処理するために、一般にパイプライン処理機能を有している。この種ベクトル演算処理装置の演算単位ごとの基本構成を第2図に示す。同図において、11は第1演算部、12はパイプラインレジスタ(P.R.)、13は第2演算部である。第2図のベクトル演算処理装置では、パイプラインレジスタ12により、演算を上下2段に分割し、それぞれを(即ち第1演算部11、および第2演算部13での各処理を)並列に動作可能とするこ

とにより、演算パイプライン処理が行なわれる構成となっている。

しかし、第2図のベクトル演算処理装置を用いて、第1図のフローチャートで示されるマスク付ベクトル演算を実行する場合、マスクビットの判断動作や、演算を実行せずにオペランド・データをそのまま出力する動作によって演算パイプラインが乱される問題があった。このため、従来のベクトル演算処理装置では、マスク付ベクトル演算の場合に演算パイプライン処理が適用できない欠点があった。

〔発明の目的〕

この発明は上記事情に鑑みてなされたものでその目的は、マスク付ベクトル演算実行においても演算パイプライン処理が適用できるベクトル演算処理装置を提供することにある。

〔発明の概要〕

この発明によれば、第1のアレイ・オペランド・データと第2のアレイ・オペランド・データ間の対応する要素毎の演算を、 M 段のパイプ

ライン処理により実行する演算部を備えたマイクロプログラム制御方式のベクトル演算処理装置が提供されている。

上記ベクトル演算処理装置では、マスク付ベクトル演算に際し、マスクビット列がシフトレジスタに初期設定される。また、マスク付ベクトル演算が開始されると、第1のアレイ・オペランド・データの各要素が、上記演算部のパイプライン処理に同期して1要素ずつバッファレジスタに順次保持され、その都度その保持要素は次段に出力される。更に、シフトレジスタは初期設定されたマスクビット列を上記演算部のパイプライン処理に同期して1ビットずつシフトする。バッファレジスタから順に出力される上記要素、およびシフトレジスタの所定位置から順に出力されるマスクビットは、上記演算部のパイプライン処理に同期して、縦続 $M-1$ 段構成の各パイプラインレジスタを順に介して出力される。選択出力手段は、最終段のパイプラインレジスタからの出力データ中の要素、また

は演算部の演算結果のいずれか一方を、マスク付ベクトル演算の演算結果の1要素として、最終段のパイプラインレジスタからの出力データ中のマスクビットに応じて選択出力する。

〔発明の実施例〕

第3図はこの発明の一実施例に係るベクトル演算処理装置の構成を示す。同図において、20はマイクロプログラム制御部、21はマイクロ制御部20からのマイクロ命令の転送路であるマイクロ命令バス(以下、M1バスと称す)である。22は第1オペランドの転送路であるデータバス(Aバスと称す)、23は第2オペランドの転送路であるデータバス(Bバスと称す)、24は演算結果の転送路であるデータバス(B2バスと称す)である。A1~Amは2段の演算パイプライン処理機能を有する演算部、Mはマスク付ベクトル制御部である。演算部A1~Amは、固有の演算機能(例えば加算、乗算など)を有している。演算部A1($i=1\sim m$)において、211はAバス22経由で導かれる第1オペランドを

保持するバッファレジスタ(BR)、321はBバス23經由で導かれる第2オペランドを保持するバッファレジスタ(BR)である。331は第1演算部(初段演算部)、341はパイプラインレジスタ(PR)、351は第2演算部(最終段演算部)である。361はマスク付ベクトル制御部Mからの後述する出力制御信号56が導かれるインバータ、371はM1バス21經由で導かれる(マイクロプログラム制御部20からの)マイクロ命令に従って演算部A1内の各部を制御する制御回路(CNT)である。381は制御回路371からの出力制御信号391およびインバータ361からの出力信号が導かれるアンドゲート、401は出力ドライバである。出力ドライバ401は、アンドゲート381からの出力信号である出力制御信号411に応じ、第2演算部351の演算結果をSバス24に出力する。

マスク付ベクトル制御部Mにおいて、51はAバス22經由で導かれる第1オペランドを保

持するバッファレジスタ(BR)、52はBバス23經由で導かれるマスクビット列が初期設定されるシフトレジスタ(SR)、53はパイプラインレジスタ(PR)である。パイプラインレジスタ53には、バッファレジスタ51からの出力データ(第1オペランド)、およびシフトレジスタ52の所定位置、例えば最上位ビット位置からの出力ビット(マスクビット)が、演算部A1の演算パイプライン処理に同期して保持される。54はM1バス21經由で導かれる(マイクロプログラム制御部20からの)マイクロ命令に従ってマスク付ベクトル制御部M内の上記各部を制御する制御回路(CNT)、55は出力ドライバである。出力ドライバ55は、パイプラインレジスタ53に保持された上記マスクビットで生成される出力制御信号56に応じ、パイプラインレジスタ53に保持された上記第1オペランドをSバス24に出力する。

次に、この発明の一実施例の動作を第4図のタイミングチャートを参照して説明する。演算

部A₁~A_mは独立に動作可能であり、マイクロプログラム制御部20からM1バス21經由で転送されるマイクロ命令によって制御される。演算部A₁~A_mは、それぞれに割当てられているマイクロ命令によって起動される。ここで、例えば演算部A₁によって処理される演算のマスク付ベクトル演算の動作を説明する。マスク付ベクトル演算においては、まずマスクビット列(M₀, M₁, ..., M_n)を、Bバス23からマスク付ベクトル制御部M内のシフトレジスタ(SR)52に取込む処理が行なわれる。次に、第4図のタイミングチャートに示される演算が行なわれる。第4図において、Kは第1演算部331の動作、Lは第2演算部351の動作を示す。またSは(マスク付ベクトル制御部Mにおいて)バッファレジスタ(BR)51からの出力データおよびシフトレジスタ(SR)52の最上位ビット位置からの出力ビットをパイプラインレジスタ(PR)53に取込むまでのタイミング、Tはパイプラインレジスタ53からSバス24へ結果を出力

するまでのタイミングを示す。

今、或るマイクロ命令によって演算部A₁にマスク付ベクトル演算の起動がかけられたものとする。このとき、同じマイクロ命令によって、マスク付ベクトル制御部Mにも起動がかけられる。演算部A₁では、制御回路371の制御により、第1のアレイ・オペランド・データ(X₀, X₁, ..., X_n)の先頭要素である第1オペランドX₀がAバス22からバッファレジスタ311に取込まれると共に、第2のアレイ・オペランド・データ(Y₀, Y₁, ..., Y_n)の先頭要素である第2オペランドY₀がBバス23からバッファレジスタ321に取込まれる。そして、バッファレジスタ311, 321に取込まれたX₀, Y₀間の演算が第1演算部331で開始される。これが前記した動作K(第4図参照)である。一方、マスク付ベクトル制御部Mでは、上記第1オペランドX₀がAバス22からバッファレジスタ51に取込まれ、前記した動作S(第4図参照)が開始される。

次のサイクルにおいて、演算部 A_1 では、第1演算部33₁からの X_0, Y_0 に関する演算の中間結果がパイプラインレジスタ34₁に取込まれる。そして、パイプラインレジスタ34₁に取込まれた中間結果に基づいて X_0, Y_0 の最終演算結果 Z_0 を生成する演算が第2演算部35₁で行なわれる。これが前記した動作L(第4図参照)である。また、演算部 A_1 では、この動作Lと並行して、次の演算対象数であるオペランド X_1, Y_1 をバス22, 23からバッファ31₁, 32₁に取込み X_1, Y_1 間の演算を開始する動作K(第4図参照)が行なわれる。一方、マスク付ベクトル制御部Mでは、バッファレジスタ51から出力される第1オペランド(この例では X_0)、およびシフトレジスタ52から出力されるマスクビット(この例ではマスクビット列の先頭ビット M_0)をパイプラインレジスタ53に取込む動作T(第4図参照)が行なわれる。また、マスク付ベクトル制御部Mでは、この動作Tと並行して、次のオペランド(第1オペランド) X_1 をAバ

ス22からバッファ51に取込むと共に、シフトレジスタ52を左1ビットシフトする動作S(第4図参照)が行なわれる。これにより、シフトレジスタ52の最上位ビット位置からは、マスクビット M_1 が出力される。なお、第4図において、符号 Δ は、シフトレジスタ52のシフトタイミングを示す。

このように、この実施例では、演算部 A_1 は第1オペランドと第2オペランドとの間の所定の演算を、マスク付ベクトル演算指定に無関係に(即ち、マスクビットの状態に無関係に)、通常のベクトル演算と同様に2段の演算パイプライン処理で実行する。また、マスク付ベクトル制御部Mは、演算部 A_1 でのパイプライン処理に同期して、2段のパイプライン処理で第1オペランドを順に取込み出力する。

マスク付ベクトル制御部Mでは、前記動作Tにおいて、パイプラインレジスタ53に取込まれているマスクビット(この例では M_0)が、当該マスク付ベクトル制御部M、および演算部 A_1

(演算部 $A_1 \sim A_m$)からのデータ出力の出力制御信号56として用いられる。この信号56は、マスク付ベクトル制御部M内の出力ドライバ55に供給されると共に、演算部 $A_1 \sim A_m$ 内のインバータ36₁~36_mにも供給される。インバータ36₁~36_mからの出力信号は、演算部 $A_1 \sim A_m$ 内の制御回路37₁~37_mからの出力制御信号39₁~39_mと共に対応するアンドゲート38₁~38_mに供給される。アンドゲート38₁~38_mからの出力信号である出力制御信号41₁~41_mは対応する出力ドライバ40₁~40_mに供給される。演算部 A_1 が起動されたこの例では、制御回路37₁~37_mからの出力制御信号39₁~39_mのうち、信号39₁だけが真("1")である。したがって、演算部 A_1 以外の演算部からの8バス24へのデータ出力は、マスク付ベクトル制御部Mからの出力制御信号56(即ちマスクビット)に無関係に禁止される。この場合、パイプラインレジスタ53からの出力制御信号56(この例ではマスクビット M_0)が偽("0")であれば、

アンドゲート38₁からの出力制御信号41₁は真("1")となり、出力ドライバ40₁は出力イネーブル状態となる。一方、マスク付ベクトル制御部M内の出力ドライバ55は、出力ディスエーブル(出力ハイ・インピーダンス)状態となる。この結果、第2演算部35₁の演算結果、即ち演算部 A_1 の演算結果(この例では Z_0)が8バス24に出力される。これに対し、パイプラインレジスタ53からの出力制御信号56(マスクビット M_0)が真("1")であれば、アンドゲート38₁からの出力制御信号41₁は偽("0")となり、出力ドライバ40₁は出力ディスエーブル状態となる。一方、マスク付ベクトル制御部M内の出力ドライバ55は出力イネーブル状態となる。この結果、パイプラインレジスタ53からの出力データ中の第1オペランド(この例では X_0)が8バス24に出力される。以下、同様の動作が第4図のタイミングチャートに示すように繰返される。

なお、上記の例では、マスク付ベクトル制御

部Mが、演算部A₁のマスク付ベクトル演算を起動するマイクロ命令によって起動された場合であるが、マスク付ベクトル制御部Mは、演算部A₁($i=1\sim m$)を起動するマイクロ命令によって起動される。したがって、マスク付ベクトル制御部Mはmの値(演算部の数)に無関係に1つでよい。ところで、マスク付ベクトル制御部M内のパイプラインレジスタ53は、当該マスク付ベクトル制御部Mが起動されていない場合には、クリアされていることが好ましい。これは、パイプラインレジスタ53をクリアすることにより、出力制御信号56が偽("0")となり、制御部Mからの出力を禁止できるためである。また、演算部A₁~A_mからの出力を、制御回路37₁~37_mからの出力制御信号39₁~39_mによって制御することができるためである。

なお、前記実施例では、2段の演算パイプライン処理を適用するベクトル演算処理装置について説明したが、この発明は3段以上のパイプライン処理を適用するベクトル演算処理装置に

も応用できる。この場合、演算パイプラインの段数をnとすると、マスク付ベクトル制御部において第1オペランドおよびマスクビットを保持し、その保持データを次段(次のパイプラインステージ)に転送するパイプラインレジスタの必要段数はn-1段となる。

(発明の効果)

以上詳述したようにこの発明によれば、少量のハードウェアを追加するだけでマスク付ベクトル演算実行においても演算パイプライン処理が適用でき、マスク付ベクトル演算の高速化が図れる。

4. 図面の簡単な説明

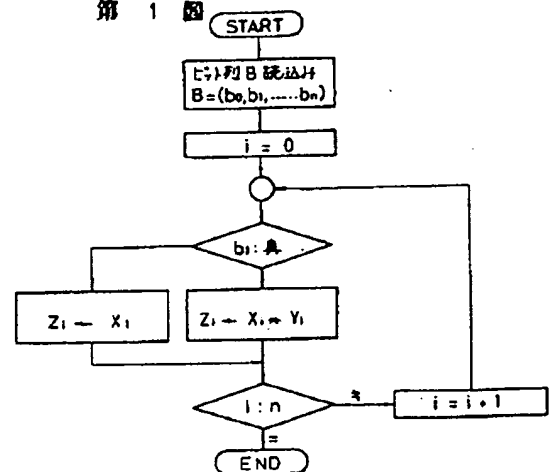
第1図は一般的なマスク付ベクトル演算を説明するフローチャート、第2図は一般的なベクトル演算処理装置の基本構成図、第3図はこの発明の一実施例に係るベクトル演算処理装置の構成図、第4図は動作を説明するためのタイミングチャートである。

A₁~A_m…演算部、M…マスク付ベクトル制御

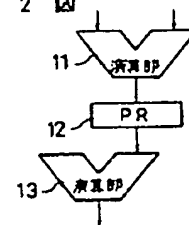
部、2.0…マイクロプログラム制御部、31₁~31_m、32₁~32_m、51…バッファレジスタ(BR)、34₁~34_m、53…パイプラインレジスタ(PR)、37₁~37_m、54…制御回路(CNT)、40₁~40_m、55…出力ドライバ、52…シフトレジスタ(SR)。

出願人代理人 弁理士 鈴 江 武 彦

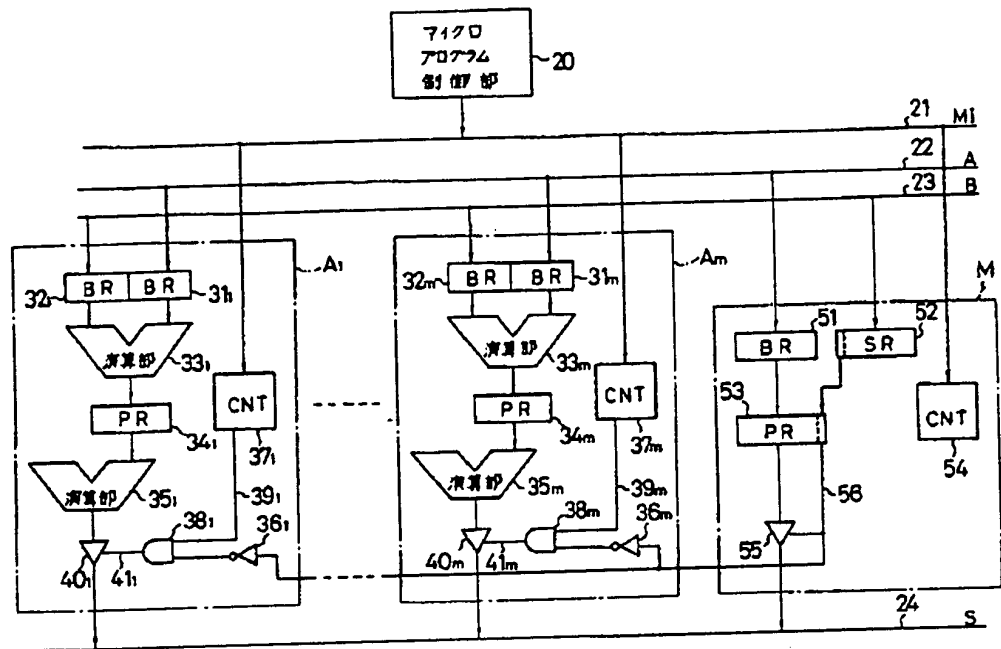
第 1 図



第 2 図



第 3 図



第 4 図

